

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-225109

(43)Date of publication of application : 03.09.1993

(51)Int.Cl.

G06F 13/10  
G06F 9/46

(21)Application number : 04-059608

(71)Applicant : NEC CORP

(22)Date of filing : 14.02.1992

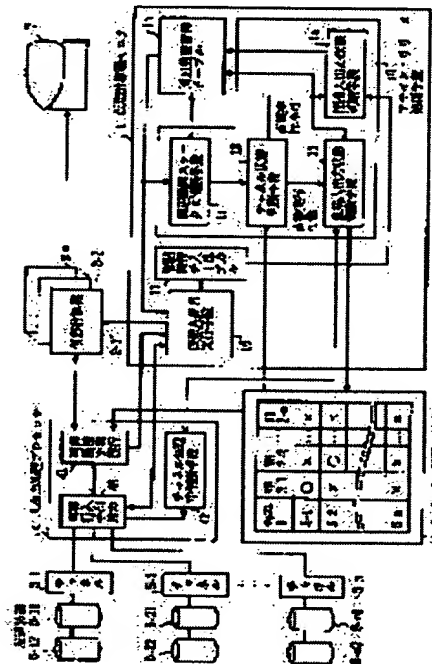
(72)Inventor : SASAKI AKITO

## (54) PERIPHERAL DEVICE SHARING SYSTEM BETWEEN VIRTUAL COMPUTERS

## (57)Abstract:

PURPOSE: To prevent the exclusive common use of peripheral devices between the virtual computers to be disabled when a virtual computer monitor releases the peripheral devices by mistake at a releasing request during the processing of an input/output request being executed directly without the intervention of the virtual computer monitor and then the peripheral devices in input/output processing is assigned to the other virtual computer.

CONSTITUTION: When a channel 5-1 is controlled by a channel control table 3 that the virtual computer 2-1 occupies, input/output requests to the peripheral devices 6-11 and 6-12 under the control of the channel 5-1 are executed directly by an input/output processing processor 4 without the intervention of the virtual computer monitor 1. When the requests to release the peripheral devices 6-11 and 6-12 are sent, the virtual computer monitor 1 itself can not decide whether or not the channel 5-1 is in processing, so an assignment releasing process means 1 checks that by the channel processing interruption judging means 42 of the input/output processing processor 4 and decides a failure in releasing when the channel is in the processing, thereby rejecting the releasing requests.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平5-225109

(43)公開日 平成5年(1993)9月3日

(51)IntCl.<sup>5</sup>

G 0 6 F 13/10  
9/46

識別記号

3 3 0 C  
3 5 0

庁内整理番号

7230-5B  
8120-5B

F I

技術表示箇所

審査請求 未請求 請求項の数 1 (全 8 頁)

(21)出願番号 特願平4-59608

(22)出願日 平成4年(1992)2月14日

(71)出願人 000004237

日本電気株式会社  
東京都港区芝五丁目7番1号

(72)発明者 佐々木 章人

東京都港区芝五丁目7番1号 日本電気株式  
会社社内

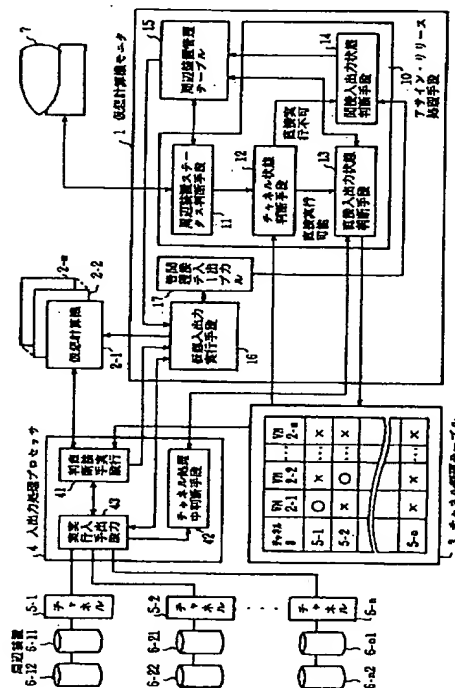
(74)代理人 弁理士 境 廣巳

(54)【発明の名称】 仮想計算機間の周辺装置共用方式

(57)【要約】

【目的】 仮想計算機モニタの介在無しに直接実行されている入出力要求の処理中に、リリース要求に応じて仮想計算機モニタが誤って周辺装置のリリースを行い、その結果、入出力処理中の周辺装置が別の仮想計算機にアサインされて、仮想計算機間の周辺装置の排他的な共用ができなくなるのを防止する。

【構成】 チャネル5-1を仮想計算機2-1が占有する旨チャネル管理テーブル3で管理されている場合、仮想計算機2-1からのチャネル5-1配下の周辺装置6-11,6-12に対する入出力要求は仮想計算機モニタ1の介在無しに直接に入出力処理プロセッサ4で実行される。周辺装置6-11,6-12のリリース要求が出されると、仮想計算機モニタ1自身ではチャネル5-1が処理中か否か判らないので、アサイン・リリース処理手段1はそのことを入出力処理プロセッサ4のチャネル処理中判断手段42で調べ、処理中であればリリース失敗としてリリース要求を拒否する。



## 【 特許請求の範囲】

【 請求項1 】 仮想計算機モニタの制御の下で動作する複数の仮想計算機からの入出力要求発行時、入出力処理プロセッサ上の直接実行判断手段が、チャンネルがどの仮想計算機に占有されているかを示す情報を保持するチャンネル管理テーブルを参照し、発行元の仮想計算機で占有中のチャンネル配下の周辺装置に対する入出力要求は仮想計算機モニタの介在無しに入出力処理プロセッサ上で直接実行し、それ以外の周辺装置に対する入出力要求は仮想計算機モニタの介在の下に間接的に実行する仮想計算機システムにおいて、

入出力処理プロセッサ上に、チェック要求の出されたチャンネルが入出力処理中か否かを要求元に通知するチャンネル処理中判断手段を備え、仮想計算機モニタ上のアサイン・リリース処理手段は、リリース要求された周辺装置の接続されたチャンネルが仮想計算機で占有中であることを前記チャンネル管理テーブルを参照して認識した場合は、前記チャンネル処理中判断手段にチェック要求を出してそのチャンネルが入出力処理中か否かを調べ、入出力処理中のときは前記リリース要求を拒否し、仮想計算機で非占有中であることを前記チャンネル管理テーブルを参照して認識した場合は、仮想計算機モニタ内部のテーブルを参照してその周辺装置に対する入出力処理中の要求があるか否かを調べ、あれば前記リリース要求を拒否することを特徴とする仮想計算機間の周辺装置共用方式。

## 【 発明の詳細な説明】

## 【 0001 】

【 産業上の利用分野】 本発明は仮想計算機モニタの制御の下で複数の仮想計算機を動作させる仮想計算機システムに関し、特に仮想計算機間の周辺装置共用方式に関する。

## 【 0002 】

【 従来の技術】 従来、この種の仮想計算機システムにおいては、周辺装置やそれを配下に持つチャンネル等の入出力資源は全ての仮想計算機で共用され得ることを想定してシステムが設計されていた。このため、各仮想計算機を制御する仮想計算機モニタにおいて、それらの入出力資源を一元的に管理、制御する構成を採用し、各仮想計算機が発行した入出力要求は、その全てを一旦仮想計算機モニタに認識させ、この仮想計算機モニタにその入出力要求の擬似等を行わせていた。

【 0003 】 しかしながら、このような仮想計算機システムでは、仮想計算機が発行した入出力要求は全て仮想計算機モニタを介して処理されるので、各々の入出力要求の実行時間に仮想計算機モニタの動作時間がオーバーヘッド時間として加わり、性能を低下させる。

【 0004 】 そこで、仮想計算機モニタの介在無しに入出力要求を処理することができるようになるために、或るチャンネルを或る仮想計算機に占有させ、その仮想計算

機からそのチャンネルに接続された周辺装置に対する入出力要求が出された場合、仮想計算機モニタに通知することなくその入出力要求を直接実行し、占有しないチャンネルに接続された周辺装置に対する入出力要求は従来通り一旦仮想計算機モニタに通知するようにした仮想計算機システムが提案されている（例えば、特願平1 - 1 8 4 9 6 2 号）。

【 0005 】 なお、前者のように仮想計算機モニタに通知することなく入出力要求を実行する方式を直接実行方式、後者のように仮想計算機モニタに通知して入出力要求を実行する方式を間接実行方式という。

## 【 0006 】

【 発明が解決しようとする課題】 ところで、周辺装置の有効利用等を促進するために、同一の周辺装置を複数の仮想計算機で排他的に共用する場合、即ち同一の周辺装置を複数の仮想計算機で時分割的に使用する場合、仮想計算機に対する周辺装置のリリース処理、アサイン処理を矛盾無く行う必要がある。

【 0007 】 即ち、例えば仮想計算機Aにアサインされていた周辺装置P Uを別の仮想計算機Bにアサインする場合、周辺装置P Uに入出力処理中の入出力要求がないことを確認した上で仮想計算機Aから周辺装置P Uをリリースする必要がある、若し、周辺装置P Uに入出力処理中の入出力要求があるときに仮想計算機Aから周辺装置P Uをリリースすると、その後の仮想計算機Bに対する周辺装置P Uのアサインが許可される結果、同一の周辺装置P Uに対して一時的に仮想計算機A、Bから同時に入出力が行われてしまう可能性がある。

【 0008 】 このような不正なリリースは、入出力要求全てを一旦仮想計算機モニタに通知する間接実行方式だけを採用した仮想計算機システムでは、仮想計算機モニタ自身が全ての周辺装置に対する入出力要求の処理状態を内部テーブルで管理しているので容易に抑止することができる。

【 0009 】 しかし、入出力処理の実行速度を高めるために、前述したように間接実行方式と直接実行方式とを混在させた仮想計算機システムでは、直接実行方式で実行される入出力要求を仮想計算機モニタが認識できないため、入出力処理中の周辺装置をリリースして他の仮想計算機にアサインしてしまう危険性があり、仮想計算機間で周辺装置を排他的に共用することが實際上困難になる。

【 0010 】 本発明はこのような事情に鑑みて為されたものであり、その目的は、間接実行方式、直接実行方式を混在可能とした仮想計算機システムにおいて入出力処理中の周辺装置を誤ってリリースしてしまうことを防止し、仮想計算機からの入出力の直接実行が行われる周辺装置であっても仮想計算機間で排他的に共用できるようにすることにある。

## 【 0011 】

【課題を解決するための手段】本発明は上記の目的を達成するために、仮想計算機モニタの制御の下で動作する複数の仮想計算機からの入出力要求発行時、入出力処理プロセッサ上の直接実行判断手段が、チャンネルがどの仮想計算機に占有されているかを示す情報を保持するチャンネル管理テーブルを参照し、発行元の仮想計算機で占有中のチャンネル配下の周辺装置に対する入出力要求は仮想計算機モニタの介在無しに入出力処理プロセッサ上で直接実行し、それ以外の周辺装置に対する入出力要求は仮想計算機モニタの介在の下に間接的に実行する仮想計算機システムにおいて、入出力処理プロセッサ上に、チェック要求の出されたチャンネルが入出力処理中か否かを要求元に通知するチャンネル処理中判断手段を備え、仮想計算機モニタ上のアサイン・リリース処理手段は、リリース要求された周辺装置の接続されたチャンネルが仮想計算機で占有中であることを前記チャンネル管理テーブルを参照して認識した場合は、前記チャンネル処理中判断手段にチェック要求を出してそのチャンネルが入出力処理中か否かを調べ、入出力処理中のときは前記リリース要求を拒否し、仮想計算機で非占有中であることを前記チャンネル管理テーブルを参照して認識した場合は、仮想計算機モニタ内部のテーブルを参照してその周辺装置に対する入出力処理中の要求があるか否かを調べ、あれば前記リリース要求を拒否する。

#### 【 0 0 1 2 】

【作用】リリース要求された周辺装置の接続されたチャンネルが仮想計算機で占有中である場合、仮想計算機モニタの介在無しにそのチャンネルを通じて周辺装置の入出力処理が行われている可能性があるが、仮想計算機モニタ自身ではそのことが不明なので、入出力処理を実際に実行している入出力処理プロセッサに設けられたチャンネル処理中判断手段に問い合わせ確認し、若しチャンネルが入出力処理中であればリリース要求を拒否することにより、不正なリリースを防止する。

【 0 0 1 3 】リリース要求された周辺装置の接続されたチャンネルが仮想計算機で非占有中の場合、そのチャンネルを通じた周辺装置の入出力処理は仮想計算機モニタが介在しているので、内部のテーブルを参照してその周辺装置に対する入出力処理中の要求があるか否かを調べ、あればリリース要求を拒否する。

#### 【 0 0 1 4 】

【実施例】次に本発明の実施例について図面を参照して詳細に説明する。

【 0 0 1 5 】図1を参照すると、本発明を適用した仮想計算機システムの一例は、仮想計算機モニタ（VMモニタ）1と、仮想計算機（VM）2-1～2-mと、チャンネル管理テーブル3と、入出力処理プロセッサ4と、チャンネル5-1～5-nと、周辺装置6-1 1, 6-1 2, 6-2 1, 6-2 2, …, 6-n 1, 6-n 2と、利用者端末7とを含んでいる。

【 0 0 1 6 】また、仮想計算機モニタ1は、アサイン・リリース処理手段1 0と、周辺装置管理テーブル1 5と、仮想入出力実行手段1 6と、間接入出力管理テーブル1 7を含み、入出力処理プロセッサ4は、直接実行判断手段4 1と、チャンネル処理中判断手段4 2と、実入出力実行手段4 3とを含んでいる。

【 0 0 1 7 】更に、アサイン・リリース処理手段1 0は、周辺装置ステータス判断手段1 1と、チャンネル状態判断手段1 2と、直接入出力状態判断手段1 3と、間接入出力状態判断手段1 4とを含んでいる。

【 0 0 1 8 】以下、上述のように構成された本実施例の動作および各部の詳細な機能を説明する。

【 0 0 1 9 】チャンネル管理テーブル3は、チャンネルがどの仮想計算機に占有中であることを示す情報を保持するテーブルであり、入出力処理プロセッサ4によって参照されると共に、仮想計算機モニタ1によって参照、更新される。

【 0 0 2 0 】図2はチャンネル管理テーブル3の或る時点の状態の一例を示す図であり、便宜上、占有中の場合には○の印を、非占有中の場合には×の印を付してある。即ち、例えばチャンネル5-1は仮想計算機2-1で占有中であり、チャンネル5-2は仮想計算機2-2で占有中である。ここで、一つのチャンネルは一つの仮想計算機に占有させるため、占有中を示す○の印は一つだけである。また、何れの仮想計算機にも占有されていないチャンネルについては全て×の印が付されている。今の場合、チャンネル5-nがそのようなチャンネルとする。

【 0 0 2 1 】他方、周辺装置管理テーブル1 5は、周辺装置6-1 1～6-n 2の仮想計算機2-1～2-mへのアサイン状態を管理するテーブルであり、仮想入出力実行手段1 6によって参照されると共に、アサイン・リリース処理手段1 0によって参照、更新される。

【 0 0 2 2 】図3は周辺装置管理テーブル1 5の或る時点の内容の一例を示す図であり、チャンネル5-1に接続される周辺装置6-1 1, 6-1 2が仮想計算機2-1にアサイン中、チャンネル5-2に接続される周辺装置6-2 1, 6-2 2が仮想計算機2-2にアサイン中、チャンネル5-nに接続される一つの周辺装置6-n 1が仮想計算機2-2にアサイン中、残りの周辺装置6-n 2が仮想計算機2-mにアサイン中となっている。

【 0 0 2 3 】さて、チャンネル管理テーブル3および周辺装置管理テーブル1 5が図2, 図3に示すような状態になっているときに、仮想計算機2-1～2-mから発行された入出力要求がどのように処理されるかを、幾つかの例を挙げて説明する。

【 0 0 2 4 】例1；仮想計算機2-1がチャンネル5-1に接続された周辺装置6-1 1に対する入出力要求を発行したとき

【 0 0 2 5 】仮想計算機2-1が発行した入出力要求は入出力処理プロセッサ4の直接実行判断手段4 1に通知

10

20

30

40

50

される。

【 0 0 2 6 】 直接実行判断手段4 1 は、チャネル管理テーブル3 を参照し、仮想計算機2 - 1 が発行した入出力要求にかかるチャネル5 - 1 がその仮想計算機2 - 1 で占有中であるか否かを調べ、図2 の場合は占有中なので、直接実行可能と判断し、その入出力要求を実入出力実行手段4 3 に通知する。

【 0 0 2 7 】 実入出力実行手段4 3 は通知された入出力要求を実行することにより、チャネル5 - 1 を通じて周辺装置6 - 1 1 に対し入出力処理を実行する。そして、入出力処理終了時にその旨を直接実行判断手段4 1 に通知し、直接実行判断手段4 1 はそれを要求元の仮想計算機2 - 1 に通知する。

【 0 0 2 8 】 このように、自仮想計算機が占有中のチャネルに接続された周辺装置に対する入出力要求は、仮想計算機モニタ1 の介在無しに入出力処理プロセッサ4 で直接に実行される。

【 0 0 2 9 】 例2 ; 仮想計算機2 - 2 がチャネル5 - n に接続された周辺装置6 - n 1 に対する入出力要求を発行し、仮想計算機2 - m がチャネル5 - n に接続された周辺装置6 - n 2 に対する入出力要求を発行したとき

【 0 0 3 0 】 仮想計算機2 - 2 , 2 - m が発行した各入出力要求は入出力処理プロセッサ4 の直接実行判断手段4 1 に通知され、直接実行判断手段4 1 は、チャネル管理テーブル3 を参照し、仮想計算機2 - 2 , 2 - m が発行した入出力要求にかかるチャネル5 - n がその仮想計算機2 - 2 , 2 - m で占有中であるか否かを調べ、図2 の状態では何れも非占有中なので、直接実行不可と判断し、それらの入出力要求を仮想計算機モニタ1 に通知する。

【 0 0 3 1 】 仮想計算機モニタ1 の仮想入出力実行手段1 6 は、直接実行判断手段4 1 から通知された入出力要求を受け取り、2 つの入出力要求が同一チャネル5 - n に対するものであるのでシリアル処理を行い、先ず一方の入出力要求を処理し、次いで残りの入出力要求を処理する。

【 0 0 3 2 】 例えば、最初に仮想計算機2 - 2 からの入出力要求を処理したとすると、仮想入出力実行手段1 6 は、その入出力要求で指定された周辺装置6 - n 1 が要求元の仮想計算機2 - 2 にアサイン中であるか否かを周辺装置管理テーブル1 5 で調べ、図3 の場合アサイン中なので、その入出力要求を擬似して他にエラーが無ければ入出力要求の処理を入出力処理プロセッサ4 の実入出力実行手段4 3 に依頼する。このとき、仮想入出力実行手段1 6 は、その入出力要求にかかる管理情報( チャネル5 - n を通じて周辺装置6 - n 1 に対し入出力処理中であり、同チャネル5 - n を通じた周辺装置6 - n 2 に対する入出力要求が待ち状態にある等を示す情報) を間接入出力管理テーブルに設定する。

【 0 0 3 3 】 実入出力実行手段4 3 は通知された入出力

要求を実行することにより、チャネル5 - n を通じて周辺装置6 - n 1 に対し入出力処理を実行する。そして、入出力処理終了時にその旨を仮想入出力実行手段1 6 に通知し、仮想入出力実行手段1 6 はそれを要求元の仮想計算機2 - 2 に通知すると共に間接入出力管理テーブル1 7 を更新する。

【 0 0 3 4 】 次に仮想入出力実行手段1 6 は、仮想計算機2 - m からの入出力要求を、仮想計算機2 - 2 からの入出力要求と同様に処理する。

【 0 0 3 5 】 このように、自仮想計算機が占有していないチャネルに接続された周辺装置に対する入出力要求は、仮想計算機モニタ1 の介在の下で実行される。

【 0 0 3 6 】 なお、仮想計算機2 - 2 がチャネル5 - n 配下の周辺装置6 - n 2 に対する入出力要求を発行した場合の如く、自仮想計算機に非アサイン中の周辺装置に対する入出力要求を発行した場合は、仮想入出力実行手段1 6 がそのことを周辺装置管理テーブル1 5 を参照して認識した時点で要求元の仮想計算機に対しエラーが通知される。

【 0 0 3 7 】 また、仮想計算機2 - 1 がチャネル5 - 2 配下の周辺装置6 - 2 1 に対する入出力要求を発行した場合の如く、自仮想計算機で非占有中のチャネル配下の非アサイン中の周辺装置に対し入出力要求を発行した場合は、直接実行判断手段4 1 で直接実行不可と判断されて仮想入出力実行手段1 6 に通知され、仮想入出力実行手段1 6 は非アサイン中の周辺装置に対する入出力要求であることを周辺装置管理テーブル1 5 を参照して認識し、要求元の仮想計算機に対しエラーを通知する。

【 0 0 3 8 】 次に、或る仮想計算機で使用していた周辺装置を別の仮想計算機で使させる際のアサイン処理、リリース処理を、幾つかの例を挙げて説明する。

【 0 0 3 9 】 例1 ; 仮想計算機2 - 1 が使用していた周辺装置6 - 1 1 , 6 - 1 2 を仮想計算機2 - 2 に使させる。

【 0 0 4 0 】 このような場合利用者は、利用者端末7 より、仮想計算機2 - 1 からの周辺装置6 - 1 1 , 6 - 1 2 のリリース要求を投入し、次に仮想計算機2 - 2 に対する周辺装置6 - 1 1 , 6 - 1 2 のアサイン要求を投入する。

【 0 0 4 1 】 仮想計算機2 - 1 からの周辺装置6 - 1 1 , 6 - 1 2 のリリース要求は仮想計算機モニタ1 のアサイン・リリース処理手段1 0 で受け付けられる。

【 0 0 4 2 】 アサイン・リリース処理手段1 0 は、先ず、周辺装置ステータス判断手段1 1 により、リリース要求された周辺装置6 - 1 1 , 6 - 1 2 が仮想計算機2 - 1 にアサイン中であるか否かを周辺装置管理テーブル1 5 を参照してチェックする。

【 0 0 4 3 】 非アサイン中であれば、当該リリース要求は不正な要求なので、リリース失敗として利用者端末7 にエラーを通知し、アサイン中であれば次の処理に進

む。周辺装置管理テーブル1 5 が図3 に示す状態の場合、周辺装置6 - 1 1 , 6 - 1 2 は仮想計算機2 - 1 にアサイン中なので、次の処理に進むことになる。

【 0 0 4 4 】次にアサイン・リリース処理手段1 0 は、チャンネル状態判断手段1 2 により、リリース要求された周辺装置6 - 1 1 , 6 - 1 2 が接続されているチャンネルが仮想計算機に占有中であるか否か( 即ち直接実行可能なチャンネルであるか否か) を調べ、占有中であれば直接入出力状態判断手段1 3 による処理に進み、非占有中であれば間接入出力状態判断手段1 4 による処理に進む。チャンネル管理テーブル3 が図2 に示す状態の場合、周辺装置6 - 1 1 , 6 - 1 2 が接続されたチャンネル5 - 1 は仮想計算機2 - 1 に占有中なので、直接入出力状態判断手段1 3 による処理に進むことになる。なお、周辺装置の接続されるチャンネルが何れのチャンネルであるかは図示しない構成テーブル等を参照することにより認識できる。

【 0 0 4 5 】直接入出力状態判断手段1 3 では、まず、チャンネル管理テーブル3 を更新してチャンネル5 - 1 の仮想計算機2 - 1 による占有を解き、新たな入出力の直接実行が行われるのを抑止する。そして、入出力処理プロセス4 のチャンネル処理中判断手段4 2 にチャンネル5 - 1 が入出力処理中か否かのチェックを要求する。

【 0 0 4 6 】チャンネル処理中判断手段4 2 は、このチェック要求に応答して、実入出力実行手段4 3 の内部テーブル等を参照することによりチャンネル5 - 1 が入出力処理中( 使用中) か否かを調べ、その結果を要求元の直接入出力状態判断手段1 3 に通知する。

【 0 0 4 7 】次に直接入出力状態判断手段1 3 によりチャンネル処理中判断手段4 2 からのチェック結果を判別し、チャンネル5 - 1 が入出力処理中であればチャンネル管理テーブル3 の状態を元に戻して即ちチャンネル5 - 1 を仮想計算機2 - 1 が占有中であるようにして、今回の周辺装置6 - 1 1 , 6 - 1 2 に対するリリース処理を失敗とする。このときアサイン・リリース処理手段1 0 は利用者端末7 にエラーを通知する。

【 0 0 4 8 】他方、チャンネル5 - 1 が入出力処理中でなければ、直接入出力状態判断手段1 3 により、周辺装置管理テーブル1 5 を更新し、周辺装置6 - 1 1 , 6 - 1 2 をリリースされた状態すなわち非アサイン中とする。そして、アサイン・リリース処理手段1 0 はリリース処理を終了する。

【 0 0 4 9 】次に、周辺装置6 - 1 1 , 6 - 1 2 の仮想計算機2 - 2 に対するアサイン処理を説明する。

【 0 0 5 0 】仮想計算機2 - 2 に対する周辺装置6 - 1 1 , 6 - 1 2 のアサイン要求は仮想計算機モニタ1 のアサイン・リリース処理手段1 0 で受け付けられる。

【 0 0 5 1 】アサイン・リリース処理手段1 0 は、まず、周辺装置ステータス判断手段1 1 により、アサイン要求された周辺装置6 - 1 1 , 6 - 1 2 のアサイン状態

を周辺装置管理テーブル1 5 を参照してチェックする。

【 0 0 5 2 】周辺装置6 - 1 1 , 6 - 1 2 がアサイン中であれば、当該アサイン要求は不正な要求なので、アサイン失敗として利用者端末7 にエラーを通知し、非アサイン中であれば次の処理に進む。

【 0 0 5 3 】従って、前述した仮想計算機2 - 1 からの周辺装置6 - 1 1 , 6 - 1 2 のリリース要求が、チャンネル5 - 1 の入出力処理中のために失敗に終わった場合には周辺装置管理テーブル1 5 は周辺装置6 - 1 1 , 6 - 1 2 をアサイン中としたままなので、上記のアサイン要求は失敗に終わることになる。

【 0 0 5 4 】他方、前述したリリース要求が正常に処理された場合、周辺装置管理テーブル1 5 が更新されて周辺装置6 - 1 1 , 6 - 1 2 は非アサイン中となっているので、アサイン・リリース処理手段1 0 は、以下の処理を行う。

【 0 0 5 5 】アサイン要求の付随情報として、直接実行モードが指定されている場合は、直接入出力状態判断手段1 3 により、チャンネル管理テーブル3 を更新して周辺装置6 - 1 1 , 6 - 1 2 が接続されているチャンネル5 - 1 を仮想計算機2 - 2 に占有させると共に、周辺装置管理テーブル1 5 を更新して周辺装置6 - 1 1 , 6 - 1 2 を仮想計算機2 - 2 にアサインされている状態にし、アサイン処理を終了する。但し、直接実行モード下においては周辺装置とチャンネルとを一括して割り当てるので、アサイン要求された周辺装置6 - 1 1 , 6 - 1 2 が接続されたチャンネル5 - 1 配下の他の周辺装置の一つでも他の仮想計算機にアサイン中であれば当該直接実行モードによるアサイン要求は拒否される。これは直接入出力状態判断手段1 3 が周辺装置管理テーブル1 5 を参照して判断する。

【 0 0 5 6 】また、直接実行モードが指定されていない場合は、間接入出力状態判断手段1 4 により、周辺装置管理テーブル1 5 を更新して周辺装置6 - 1 1 , 6 - 1 2 を仮想計算機2 - 2 にアサインされている状態にし、アサイン処理を終了する。

【 0 0 5 7 】例2 ; 仮想計算機2 - m が使用していた周辺装置6 - n 2 を仮想計算機2 - 2 に使用させる。

【 0 0 5 8 】このような場合利用者は、利用者端末7 より、仮想計算機2 - m からの周辺装置6 - n 2 のリリース要求を投入し、次に仮想計算機2 - 2 に対する周辺装置6 - n 2 のアサイン要求を投入する。

【 0 0 5 9 】仮想計算機2 - m からの周辺装置6 - n 2 のリリース要求は仮想計算機モニタ1 のアサイン・リリース処理手段1 0 で受け付けられ、アサイン・リリース処理手段1 0 は、まず、周辺装置ステータス判断手段1 1 により、リリース要求された周辺装置6 - n 2 が仮想計算機2 - m にアサイン中であるか否かを周辺装置管理テーブル1 5 を参照してチェックし、非アサイン中であれば、リリース失敗として利用者端末7 にエラーを通知

し、アサイン中であれば次の処理に進む。周辺装置管理テーブル1 5 が図3 に示す状態の場合、周辺装置6 - n 2 は仮想計算機2 - m にアサイン中なので、次の処理に進む。

【 0 0 6 0 】次にアサイン・リリース処理手段1 0 は、チャンネル状態判断手段1 2 により、リリース要求された周辺装置6 - n 2 が接続されているチャンネル5 - n が仮想計算機に占有中であるか否かを調べ、図2 に示すチャンネル管理テーブル3 の場合、周辺装置6 - n 2 が接続されたチャンネル5 - n は何れの仮想計算機2 - m にも占有

【 0 0 6 1 】間接入出力状態判断手段1 4 では、まず、間接入出力管理テーブル1 7 を参照し、周辺装置6 - n 2 について入出力処理中の要求があるか否かを調べ、入出力処理中の要求があった場合はリリース失敗として処理を終了する。

【 0 0 6 2 】他方、入出力処理中の要求がなかった場合、間接入出力状態判断手段1 4 により、周辺装置管理テーブル1 5 を更新して周辺装置6 - n 2 を仮想計算機2 - m からリリースされた状態にし、処理を終了する。

【 0 0 6 3 】次に、周辺装置6 - n 2 の仮想計算機2 - 2 に対するアサイン処理を説明する。

【 0 0 6 4 】仮想計算機2 - 2 に対する周辺装置6 - n 2 のアサイン要求を受け付けた仮想計算機モニタ1 のアサイン・リリース処理手段1 0 は、まず、周辺装置ステータス判断手段1 1 により、アサイン要求された周辺装置6 - n 2 のアサイン状態を周辺装置管理テーブル1 5 を参照してチェックする。

【 0 0 6 5 】周辺装置6 - n 2 がアサイン中であれば、アサイン失敗として利用者端末7 にエラーを通知し、非アサイン中であれば次の処理に進む。

【 0 0 6 6 】従って、前述した仮想計算機2 - m からの周辺装置6 - n 2 のリリース要求が、周辺装置6 - n 2 の入出力処理中のために失敗に終わった場合には周辺装置管理テーブル1 5 は周辺装置6 - n 2 をアサイン中としたままなので、上記のアサイン要求は失敗に終わることになる。

【 0 0 6 7 】他方、前述したリリース要求が正常に処理された場合、周辺装置管理テーブル1 5 が更新されて周辺装置6 - n 2 は非アサイン中となっているので、アサイン・リリース処理手段1 0 は、以下の処理を行う。

【 0 0 6 8 】アサイン要求の付随情報として、直接実行モードが指定されている場合は、直接入出力状態判断手段1 3 により、チャンネル管理テーブル3 を更新して周辺装置6 - n 2 が接続されているチャンネル5 - n を仮想計算機2 - 2 に占有させると共に、周辺装置管理テーブル1 5 を更新して周辺装置6 - n 2 を仮想計算機2 - 2 にアサインされている状態にし、アサイン処理を終了する。但し、アサイン要求された周辺装置6 - n 2 が接続

されたチャンネル5 - n 配下の他の周辺装置の一つでも他の仮想計算機にアサイン中であれば当該直接実行モードによるアサイン要求は拒否される。これは直接入出力状態判断手段1 3 が周辺装置管理テーブル1 5 を参照して判断する。

【 0 0 6 9 】また、直接実行モードが指定されていない場合は、間接入出力状態判断手段1 4 により、周辺装置管理テーブル1 5 を更新して周辺装置6 - n 2 を仮想計算機2 - 2 にアサインされている状態にし、アサイン処理を終了する。

【 0 0 7 0 】

【 発明の効果 】以上説明したように、本発明は、チャンネルが処理中か否かを調べて要求元に通知するチャンネル処理中判断手段を入出力処理プロセッサ上に設け、仮想計算機モニタ上のアサイン・リリース処理手段は、或る仮想計算機から或る周辺装置をリリースする要求を受けたとき、その周辺装置が接続されたチャンネルが仮想計算機に占有中であることをチャンネル管理テーブルで認識すると、チャンネル処理中判断手段に要求を出してそのチャンネルが入出力処理中か否かを調べ、入出力処理中であればリリース要求を拒否し、またその周辺装置が接続されたチャンネルが仮想計算機に占有されていないことを認識すると、仮想計算機モニタの内部テーブルである間接入出力管理テーブルを参照してその周辺装置に対する入出力処理中の要求があるか否かを調べ、あればリリース要求を拒否するようにしたので、仮想計算機からの入出力の間接実行が行われる周辺装置はもとより、仮想計算機からの入出力の直接実行が行われる周辺装置であっても、入出力処理中の周辺装置を誤ってリリースしてしまうことが防止できる。

【 0 0 7 1 】これによって、同一の周辺装置に対し複数の仮想計算機から同時に入出力が行われることを防止でき、間接実行方式、直接実行方式の混在する仮想計算機システムにおいて仮想計算機間の周辺装置の排他的な共用が実現できる。

【 図面の簡単な説明 】

【 図1 】本発明の仮想計算機間の周辺装置共用方式を適用した仮想計算機システムの一例を示すブロック図である。

【 図2 】チャンネル管理テーブルの或る時点の状態の一例を示す図である。

【 図3 】周辺装置管理テーブルの或る時点の内容の一例を示す図である。

【 符号の説明 】

1 …仮想計算機モニタ ( VMモニタ )

1 0 …アサイン・リリース処理手段

1 1 …周辺装置ステータス判断手段

1 2 …チャンネル状態判断手段

1 3 …直接入出力状態判断手段

1 4 …間接入出力状態判断手段

11

12

1 5 …周辺装置管理テーブル  
 1 6 …仮想入出力実行手段  
 1 7 …間接入出力管理テーブル  
 2 - 1 ~ 2 - m …仮想計算機 ( VM )  
 3 …チャネル管理テーブル  
 4 …入出力処理プロセッサ

4 1 …直接実行判断手段  
 4 2 …チャネル処理中判断手段  
 4 3 …実入出力実行手段  
 5 - 1 ~ 5 - n …チャネル  
 6 - 1 1 ~ 6 - n 2 …周辺装置  
 7 …利用者端末

【 図2 】

仮想計算機 チャネル	2 - 1	2 - 2	...	2 - m
5 - 1	○	×	...	×
5 - 2	×	○	...	×
⋮	⋮	⋮		⋮
5 - n	×	×	...	×

【 図3 】

周辺装置	アサイン先仮想計算機
6 - 1 1	2 - 1
6 - 1 2	2 - 1
6 - 2 1	2 - 2
6 - 2 2	2 - 2
⋮	⋮
6 - n 1	2 - 2
6 - n 2	2 - m



【 図1 】

